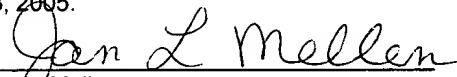
**PRE-APPEAL BRIEF REQUEST FOR REVIEW**

Docket No. SMY-041.01

Applicant: Nir N. Shavit, Steven K. Heller and Christine H. Flood
Serial No: 09/893,256
Filed: June 27, 2001
For: LOAD-BALANCING QUEUES EMPLOYING LIFO/FIFO WORK STEALING
Examiner: C. Truong
Art Unit: 2195

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

The undersigned hereby certifies that this document is being placed in the United States mail with first-class postage attached, addressed to Mail Stop AF, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450 on December 23, 2005.


Jan L. Mellen

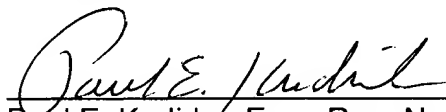
Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Applicant requests review of the rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal

The review is requested for the reason(s) stated on that attached sheet(s).

Respectfully submitted



Date: 12/23/05

Paul E. Kudirka, Esq. Reg. No. 26,931
KUDIRKA & JOBSE, LLP
Customer Number 045774
Tel: (617) 367-4600 Fax: (617) 367-4656

REASONS FOR REVIEW

Claims 1-47 have been rejected under 35 U.S.C. §112, second paragraph, as indefinite for using the phrase “popping a task identifier from one of the top end and the bottom end of that task queue...” This phrase is used in the independent claims 1, 10, 15, 24, 33 and 42 and the remainder of the claims have been rejected as dependent on a rejected base claim. Applicants’ attorney conducted a telephone conference with the examiner where this rejection was discussed. During that conference, the examiner indicated that the wording “from one of the top end and the bottom end of that task queue” is not clear because a task identifier cannot be simultaneously popped off both the top and the bottom end of the queue. Consequently, applicant’s attorney and the examiner agreed that the wording should be “popping a task identifier from one of the top end or the bottom end of that task queue...” Claims 1, 10, 15, 24, 33 and 42 were amended in the response filed July 14, 2005 to replace the “and” to which the examiner has objected with an “or.” Consequently, this rejection is respectfully traversed.

Claims 1-4, 8, 15-18, 22, 24-27, 31, 33-36, 40 and 42 have been rejected under 35 U.S.C. §103(a) as obvious over U.S. Patent No. 5,410,722 (Cornaby) in view of U.S. Patent No. 4,482,956 (Tallman.) Applicant contends that the examiner has failed to establish a *prima facie* basis for rejection the claims as obvious because neither cited reference discloses the claimed limitations.

The examiner asserts that Cornaby discloses all of the recited limitations with the exception that Cornaby does not explicitly disclose a mode-selection criterion. However, the examiner asserts that a mode-selection criterion is disclosed in Tallman and concludes that it would have been obvious to combine the teachings of Cornaby and Tallman in order to increase the flexibility of Cornaby’s system.

As discussed in the response filed December 24, 2004, the Cornaby reference discloses a queue system in which there are multiple task queues, each of which is processed in a single manner, such as a FIFO manner, a LIFO manner or another manner. The processing manner for a particular task queue is chosen by the system designer at the time that the system is designed. In order to change the manner in which a task is processed, a register for the task is moved from one queue to another.

Thus, to change the processing manner of a task that is in a LIFO task queue, to a FIFO processing manner, the task is moved to a queue that is being processed in a FIFO manner.

The Tallman patent discloses several manners for processing queue elements. In the section to which the examiner refers (Tallman, column 5, line 54-column 6, line 6) the Tallman patent discloses the use of two queues: a LIFO queue and a FIFO queue. The process starts with the first queue being processed in a LIFO manner. When certain mode criteria are met (such as the LIFO queue reaching a maximum size or the FIFO queue becoming empty) the elements in the LIFO queue are removed, reordered in FIFO order and transferred to the FIFO queue (See Tallman, column 5, line 66 – column 6, line 6. The other element processing arrangement allows a single queue to be operated in a LIFO manner by one program and a FIFO manner by another program. See, for example, Tallman, column 6, lines 34-61. The choice of whether to process the queue in a LIFO manner or a FIFO manner is left to the user's choice (Tallman, column 6, lines 58-61.) Tallman does not disclose using different programs to process the same queue in a LIFO manner or a FIFO manner based on mode-changing criteria, such as queue length.

Neither reference discloses or suggests a single thread changing the processing mode of a single queue on the basis of mode criteria. In particular, in accordance with the principles of the present invention, the processing of each task queue by the one thread can be dynamically switched between FIFO mode and LIFO mode by removing task identifiers from one end or the other end of the same queue. As mentioned above, Cornaby discloses changing modes by changing queues. Tallman also discloses changing modes by changing queues and also discloses changing modes on the same queue by using different programs. Therefore, the combination of Cornaby and Tallman, as suggested by the examiner, would either suggest changing queues (which both references disclose) or using different programs to manipulate the same queue as Tallman discloses. Nothing in Tallman would have suggested modifying the multiple queue arrangement of Cornaby to use a single queue. Similarly, nothing in Cornaby would have suggested modifying Tallman to use the same program to process the same queue in two different manners.

Independent claims 1, 15, 24, 33 and 42 clearly recite this difference. Claim 1 is illustrative and recites that a separate execution thread is employed for each provided task queue to repeatedly pop a task identifier from either the top end or the bottom end of that task queue in order to access that task queue in either a LIFO manner or a FIFO manner (emphasis added). See claim 1, lines 5-15. As previously mentioned, the combination of Cornaby and Tallman does not disclose or suggest that type of operation. Consequently, claim 1 distinguishes over the cited reference. Independent claims 15, 24, 33, and 42 contain similar wording and distinguish over the cited reference in same manner as claim 1.

Claims 2-4 and 8 are dependent, either directly or indirectly on claim 1 and incorporate the limitations thereof. Therefore, they distinguish over the cited combination of Cornaby and Tallman in the same manner as claim 1. In addition, these claims recite further elements not shown or disclosed by Cornaby or Tallman or suggested by the combination. For example, claim 8 recites that the execution thread for a task queue that contains no entries can perform tasks from another task queue. As discussed in the response to the last office action, Cornaby does not disclose or suggest that the queue handler for an empty queue could or should process other tasks if the empty queue contains no task registers as recited in claim 8. Tallman discloses that when a queue is empty, elements from another queue are inserted into that queue for processing (column 5, line 66- column 6, line 6). Thus, claim 8 patentably distinguishes over the cited Cornaby and Tallman combination.

Claims 16-18 and 22 are dependent, either directly or indirectly on claim 15 and incorporate the limitations thereof. Therefore, they distinguish over the cited Cornaby and Tallman combination in the same manner as claim 15. Further, the limitations in claims 16-18 and 22 parallel those in claims 2-4 and 8 and, consequently, the former claims distinguish over the cited reference combination in the same manner as the latter claims.

Claims 25-27 and 31 are dependent, either directly or indirectly on claim 24 and incorporate the limitations thereof. Therefore, they distinguish over the cited combination in the same manner as claim 24. Further, the limitations in claims 25-27 and 31 parallel those in claims 2-4 and 8 and, consequently, the former claims

distinguish over the cited reference combination in the same manner as the latter claims.

Claims 34-36 and 40 are dependent, either directly or indirectly on claim 33 and incorporate the limitations thereof. Therefore, they distinguish over the cited combination in the same manner as claim 33. Further, the limitations in claims 34-36 and 40 parallel those in claims 2-4 and 8 and, consequently, the former claims distinguish over the cited combination in the same manner as the latter claims.

Claims 5-7, 9-14, 19-21, 23, 28-30, 32, 37-39, 41 and 43-47 have been rejected under 35 U.S.C. §103(a) over Cornaby in view of Tallman and further in view of applicant admitted prior art (AAPA). The examiner comments that Cornaby and Tallman disclose the invention as claimed but do not explicitly teach that the tasks performed implement a garbage collection system. However, the examiner indicates the AAPA discloses that garbage collectors are well-known and that it would have been obvious to combine Cornaby, Tallman and the AAPA garbage collectors in order to provide automatic reclamation of dynamically allocated memory.

Claims 5-7 and 9 are dependent, either directly or indirectly, on claim 1 and incorporate the limitations thereof. Therefore, they distinguish over the cited Cornaby and Tallman combination in the same manner as claim 1. In addition, it is noted that neither Cornaby nor Tallman mention or discuss the problem solved by the present invention – limiting task queue size. Thus, there is no motivation for using Cornaby or Tallman with garbage collection systems. Further, even if the Cornaby system was applied to garbage collectors in order to limit queue size, in the Cornaby system this would be accomplished by shifting task between queues, not by changing the processing mode of each queue as claimed. A similar result obtains for Tallman. Consequently, claims 5-7 and 9 patentably distinguish over the Cornaby, Tallman and AAPA combination.

Claim 10 contains limitations that parallel those in claim 1 and accordingly it distinguishes over the cited reference in the same manner as claim 1. Claims 11-14 are dependent, either directly or indirectly on claim 10 and incorporate the limitations thereof. Therefore, they distinguish over the cited Cornaby/Tallman/AAPA combination in the same manner as claim 10. In addition, claim 12 contains limitations that parallel

those in claim 8 and therefore distinguishes over the cited reference combination in the same manner as claim 8 discussed above.

Claim 19-21 and 23 are dependent, either directly or indirectly on claim 15 and incorporate the limitations thereof. Therefore, they distinguish over the cited reference combination in the same manner as claim 15.

Claims 28-30 and 32 are dependent, either directly or indirectly on claim 24 and incorporate the limitations thereof. Therefore, they distinguish over the cited Cornaby, Tallman and AAPA references in the same manner as claim 24.

Claims 37-39 and 41 are dependent, either directly or indirectly on claim 33 and incorporate the limitations thereof. Therefore, they distinguish over the cited Cornaby, Tallman and AAPA references in the same manner as claim 33.

Claims 43-47 are dependent, either directly or indirectly on claim 1, 10, 15, 24 and 33 and incorporate the limitations thereof. Therefore, they distinguish over the cited Cornaby, Tallman and AAPA references in the same manner as claim 33.